


**Operating System and Design (19CS2106A)**  
**Advanced Lab- 4**

**Xv6 design, implementation, and customization.**

**Clear:**

```
#include "types.h"
#include "stat.h"
#include "user.h"
#include "fs.h"
int
main(int argc, char *argv[])
{
    printf(1, "\033[2J\033[1;1H\n");
    exit();
}
```



```
osd-190031187@team-osd:~/xv6-public
GNU nano 2.3.1 File: clear.c
#include "types.h"
#include "user.h"
void clear(int x)
{
    if (x=='x')
        return;
    printf(1, "\xa");
    clear(x+('l'-48));
}
int main(void)
{
    clear('A');
    exit();
}
```

```

osd-190031187@team-osd:~/xv6-public
SeaBIOS (version 1.11.0-2.el7)

iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+1FF94780+1FED4780 C980

Booting from Hard Disk..xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap 8
init: starting sh
190031187$ ls
.          1 1 512
..         1 1 512
README    2 2 2286
cat        2 3 14612
echo       2 4 13464
forktest   2 5 8288
grep       2 6 16148
init       2 7 14356
kill       2 8 13496
ln         2 9 13436
ls         2 10 16292
mkdir      2 11 13524
rm         2 12 13500
sh         2 13 24940
stressfs   2 14 14452
usertests  2 15 67348
wc         2 16 15272
zombie     2 17 13160
ps         2 18 14188
bt         2 19 13728
touch      2 20 13688
alarmtest  2 21 13900
touchex    2 22 13696
tail       2 23 18368
printf     2 24 12556
shutdown   2 25 13116
clear      2 26 13512
console    3 27 0
190031187$ █

```

## Shutdown

```

#include "types.h"
#include "stat.h"
#include "user.h"
int main(int argc, char *argv[])
{
    halt();
    exit();
}

```

we have to add halt system call to make it(shutdown) work

Add SYSCALL(halt) to usys.S

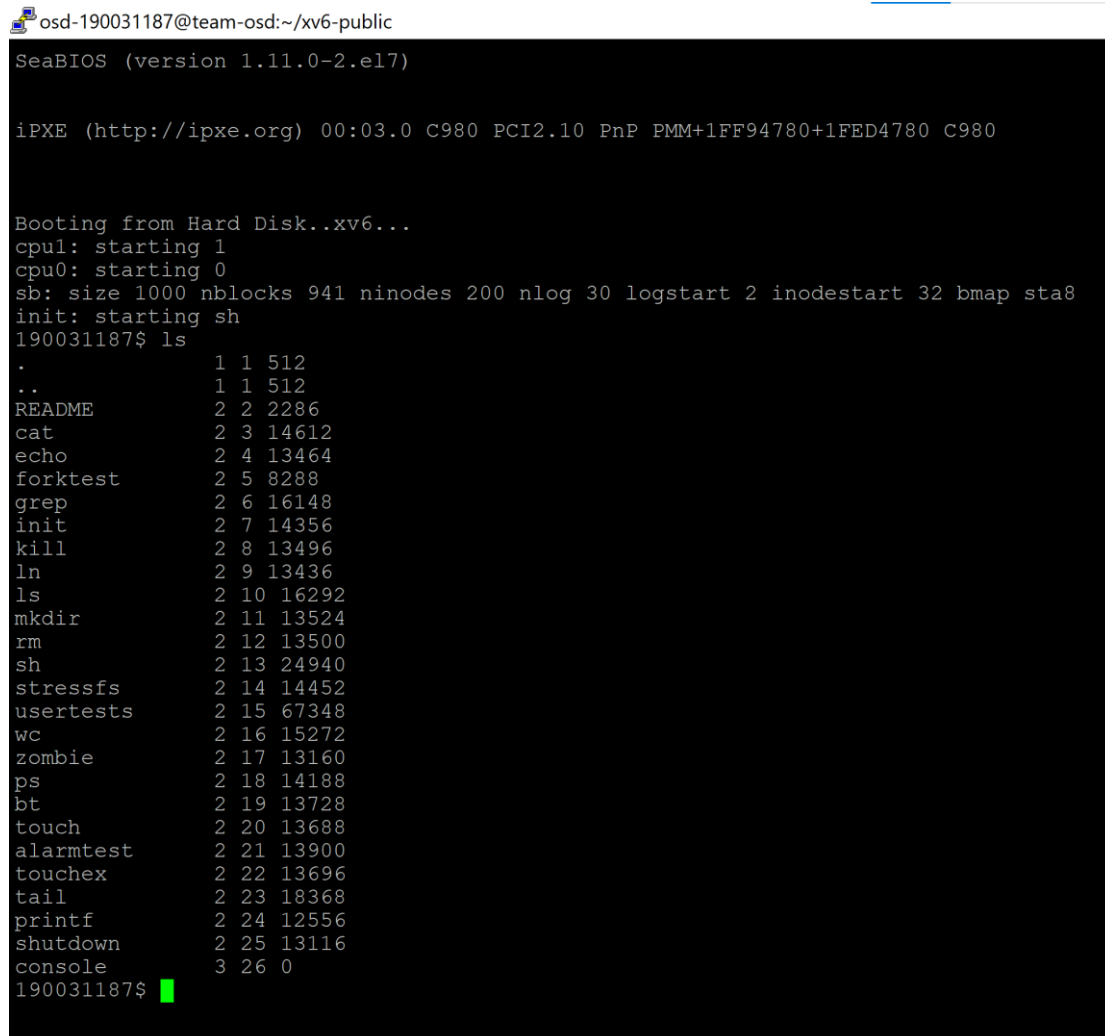
Add #define SYS\_halt 22 to syscall.h

Add extern int sys\_halt(void); and [SYS\_halt] sys\_halt, to syscall.c

Add int halt(void); to user.h

Add the following code to sysproc.c

```
int
sys_halt(void)
{
outb(0xf4, 0x00);
return 0;
}
```




```
osd-190031187@team-osd:~/xv6-public
SeaBIOS (version 1.11.0-2.el7)

iPXE (http://ipxe.org) 00:03.0 C980 PCI2.10 PnP PMM+1FF94780+1FED4780 C980

Booting from Hard Disk..xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap sta8
init: starting sh
190031187$ ls
.          1 1 512
..         1 1 512
README    2 2 2286
cat       2 3 14612
echo      2 4 13464
forktest  2 5 8288
grep      2 6 16148
init      2 7 14356
kill      2 8 13496
ln        2 9 13436
ls        2 10 16292
mkdir     2 11 13524
rm        2 12 13500
sh        2 13 24940
stressfs  2 14 14452
usertests 2 15 67348
wc        2 16 15272
zombie    2 17 13160
ps        2 18 14188
bt        2 19 13728
touch     2 20 13688
alarmtest 2 21 13900
touchex   2 22 13696
tail      2 23 18368
printf    2 24 12556
shutdown  2 25 13116
console   3 26 0
190031187$ █
```

## UNIX system programming

- 1. Link Library Shared libraries** (also called dynamic libraries) are linked into the program in two stages. First, during compile time, the link verifies that all the symbols (again, functions, variables and the like) required by the program, are either linked into the program, or in one of its shared libraries. To use a Library that is not linked into your program automatically by the compiler, you need to (1) include the library's header file in your C source file (test.c in the example below), and (2) tell the compiler to link in the code from the library .o file into your executable file

 osd-190031187@team-osd:~/xv6-public

```
[osd-190031187@team-osd xv6-public]$ nano ctest1.c
[osd-190031187@team-osd xv6-public]$ nano ctest2.c
[osd-190031187@team-osd xv6-public]$ nano prog.c
[osd-190031187@team-osd xv6-public]$ ar -cvq libctest.a ctest1.o ctest2.o
a - ctest1.o
a - ctest2.o
[osd-190031187@team-osd xv6-public]$ ./prog
Valx=5
[osd-190031187@team-osd xv6-public]$ █
```

## 2. makefile

### 1. Create file : nano hellomake.c

```
#include <hellomake.h>

int main() {
    // call a function in another file
    myPrintHelloMake();

    return(0);
}
```

### 2. Create file : nano hellofunc.c

```
#include <stdio.h>
#include <hellomake.h>
void myPrintHelloMake(void) {
    printf("Hello makefiles!\n");
    return;
}
```

### 3. Create file : nano hellomake.h

```
/*
example include file
*/

void myPrintHelloMake(void);
```

**To Compile the code execute the following command:**

```
gcc -o hellomake hellomake.c hellofunc.c -I.
```

osd-190031187@team-osd:~/xv6-public

```
[osd-190031187@team-osd xv6-public]$ cat hellomake
ELF>@@@x@8      @@@@8888@@@DD ``$( ( ( ( TT@T@DDPtd@@@<<QtdRtd``/ls
G

Ht;H5
%
@%
h%
h%
h1I^HHPHI@HPEH-@fD?`UH-8`H`HM
:
8

L T9 I \ 4zRx
*zRx
$ @FJ
K ?;*3$"D A C
Lu A C
D h eB E E E (H0 H8 M@18A0A(B BB @
o @
=
`H @ o ` o oV@(`@&@6@GCC: (GNU) 4.8.5 20150623 (Red Hat 4.8.5-39)8@
@
@ @ @ @ @ `` ` ( `` `0`4`
-@ 8 `{O4`7 @vu0` @ @ @ @ -
@crtstuff.c __JCR_LIST__deregister_tm_clones__do_global_dtors_auxcomplt
```

### 3. namedpipe: fifo

#### Reader.c

```
#include <stdio.h>
#include <sys/types.h>
#include <fcntl.h>
/*****
main ()
{
    int fd;
    char str[100];
    mkfifo ("aPipe", 0660); /* Create named pipe */
    fd = open ("aPipe", O_RDONLY); /* Open it for reading */
    while (readLine (fd, str)) /* Display received messages */
        printf ("%s\n", str);
    close (fd); /* Close pipe */
}
/*****/
readLine (fd, str)
int fd;
char* str;
/* Read a single NULL-terminated line into str from fd */
/* Return 0 when the end-of-input is reached and 1 otherwise */
{
    int n;
    do /* Read characters until NULL or end-of-input */
    {
        n = read (fd, str, 1); /* Read one character */
    }
}
```

```

while (n > 0 && *str++ != 0);
return (n > 0); /* Return false if end-of-input */
}

```

## Writer.c

```

#include <stdio.h>
#include <fcntl.h>
/*****
main ()
{
    int fd, messageLen, i;
    char message [100];
    /* Prepare message */
    sprintf (message, "Hello from PID %d", getpid ());
    messageLen = strlen (message) + 1;
    do /* Keep trying to open the file until successful */
    {
        fd = open ("aPipe", O_WRONLY); /* Open named pipe for writing */
        if (fd == -1) sleep (1); /* Try again in 1 second */
    }
    while (fd == -1);
    for (i = 1; i <= 3; i++) /* Send three messages */
    {
        write (fd, message, messageLen); /* Write message down pipe */
        sleep (3); /* Pause a while */
    }
    close (fd); /* Close pipe descriptor */
}

```

```

osd-190031187@team-osd:~
GNU nano 2.3.1      File: Reader.c
#include <stdio.h>
#include <sys/types.h>
#include <fcntl.h>
/*****
main ()
{
    int fd;
    char str[100];
    mkfifo ("aPipe", 0660); /* Create named pipe */
    fd = open ("aPipe", O_RDONLY); /* Open it for reading */
    while (readLine (fd, str)) /* Display received messages */
        printf ("%s\n", str);
    close (fd); /* Close pipe */
}
/*****
readLine (fd, str)
int fd;
char* str;
/* Read a single NULL-terminated line into str from fd */
/* Return 0 when the end-of-input is reached and 1 otherwise */
{
    int n;
    do /* Read characters until NULL or end-of-input */
    {
        n = read (fd, str, 1); /* Read one character */
    }
    while (n > 0 && *str++ != 0);
    return (n > 0); /* Return false if end-of-input */
}

```

```
osd-190031187@team-osd:~  
GNU nano 2.3.1 File: Writer.c  
#include <stdio.h>  
#include <fcntl.h>  
#include <string.h>  
/*****  
main ()  
{  
    int fd, messageLen, i;  
    char message [100];  
    /* Prepare message */  
    sprintf (message, "Hello from PID %d", getpid ());  
    messageLen = strlen (message) + 1;  
    do /* Keep trying to open the file until successful */  
    {  
        fd = open ("aPipe", O_WRONLY); /* Open named pipe for writing */  
        if (fd == -1) sleep (1); /* Try again in 1 second */  
    }  
    while (fd == -1);  
    for (i = 1; i <= 3; i++) /* Send three messages */  
    {  
        write (fd, message, messageLen); /* Write message down pipe */  
        sleep (3); /* Pause a while */  
    }  
    close (fd); /* Close pipe descriptor */  
}
```

```
osd-190031187@team-osd:~  
[osd-190031187@team-osd ~]$ ./reader & ./reader & ./writer  
[1] 58360  
[2] 58361  
Hello fro I 86  
e o D56  
mPD532HllofrmPI 832Hello from PID 58362  
[1]- Done ./reader  
[2]+ Done ./reader  
[osd-190031187@team-osd ~]$
```